




# FreeBSDで完全冗長化された Network Filesystemを

作れなかつた記録

2016/02/18

許 先明

# お品書き

-  Motivation
-  Implement
-  Problems

Confidential

# Motivation

# なんで？

- 📌 Mail Serverなどに供給するFilesystemは安定性と冗長性が重要
  - 📌 一般にFilesystemの冗長化は非常に難しい
  - 📌 Freeに利用できる良い実装はない。
  - 📌 MailServerを構築する場合
    - 📌 Lock問題への対応
      - 📌 Maildirを利用しHome Directoryに
      - 📌 ファイル単位で
      - 📌 受信したメールを保存する
    - 📌 SMTP Serverは冗長性を確保したい
      - 📌 Mail spoolの冗長化を考えると Network Filesystemを利用したくなる
- 📌 負荷分散を考える必要があるシステムで、コンテンツを共有する必要があるものには、常に存在する

# どうやって？

## 📌 Linux系(CentOS/Ubuntu…)

- 📌 DRBD : Disk単位(Block Device単位)での同期
  - 📌 Block DeviceとしてのHDDを同期
- 📌 Isyncd/rsync : ファイル単位での同期
  - 📌 Linux kernelのinotifyを利用したファイルの書き換え検出(Isyncd)
  - 📌 検出されたファイルの同期(rsync)
    - 📌 Symbolic Linkの取り扱いに難がある（というはなしをきいた）

## 📌 FreeBSD系

- 📌 HAST : Disk単位(Block Device単位)での同期

## 📌 その他

- 📌 知らない。NetBSDとOpenBSDには、このような実装はないと思われ

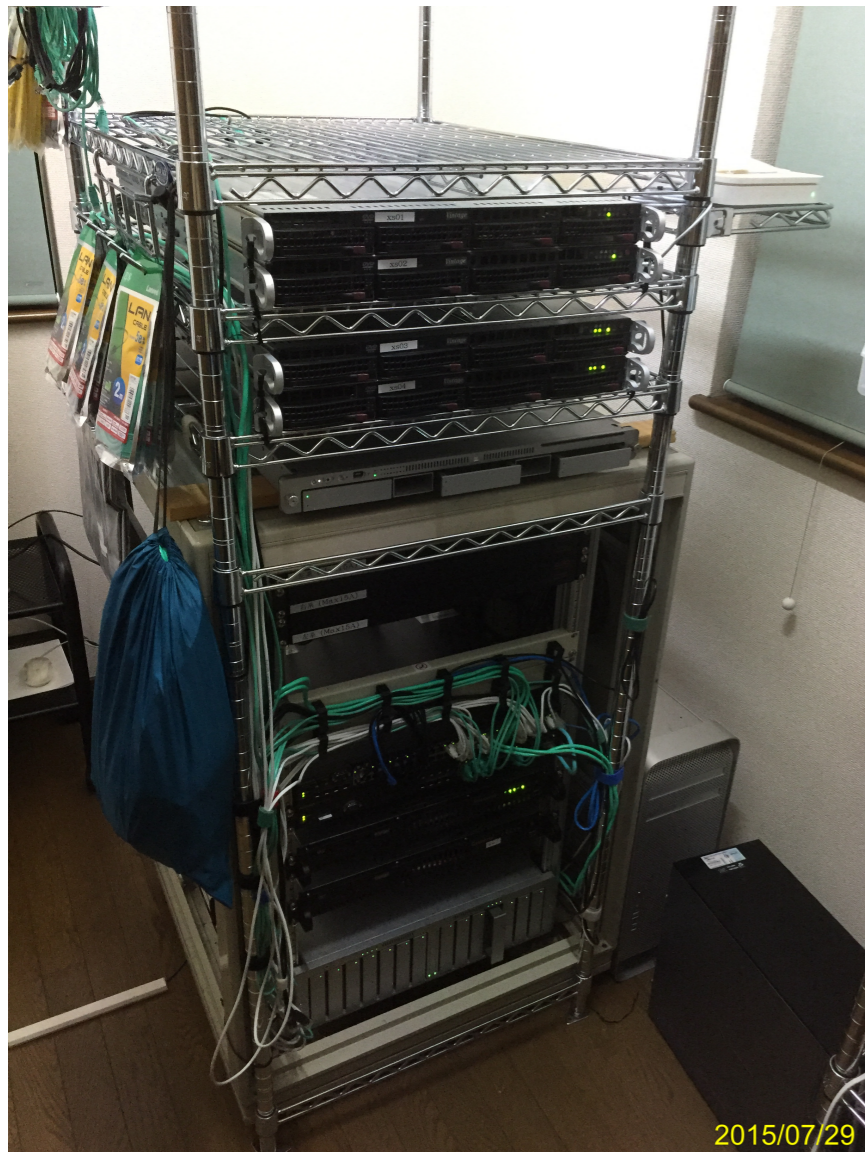
# FreeBSDでHAST?

- 📌 やっぱり運用するなら(慣れているし)\*BSDだよね
- 📌 HASTしかないんだけど。しかもFreeBSD系だけなんだけど。
- 📌 やっている人すくなそう。情報も少ないぞ
- 📌 NAS4FreeとかFreeNASとか
- 📌 どうせExperimentalだし
- 📌 GUIから(全部は)設定できないし
- 📌 情報少ないし

# Implement

# 我が家の環境 -1-

## Storages



Server  
&  
Switch

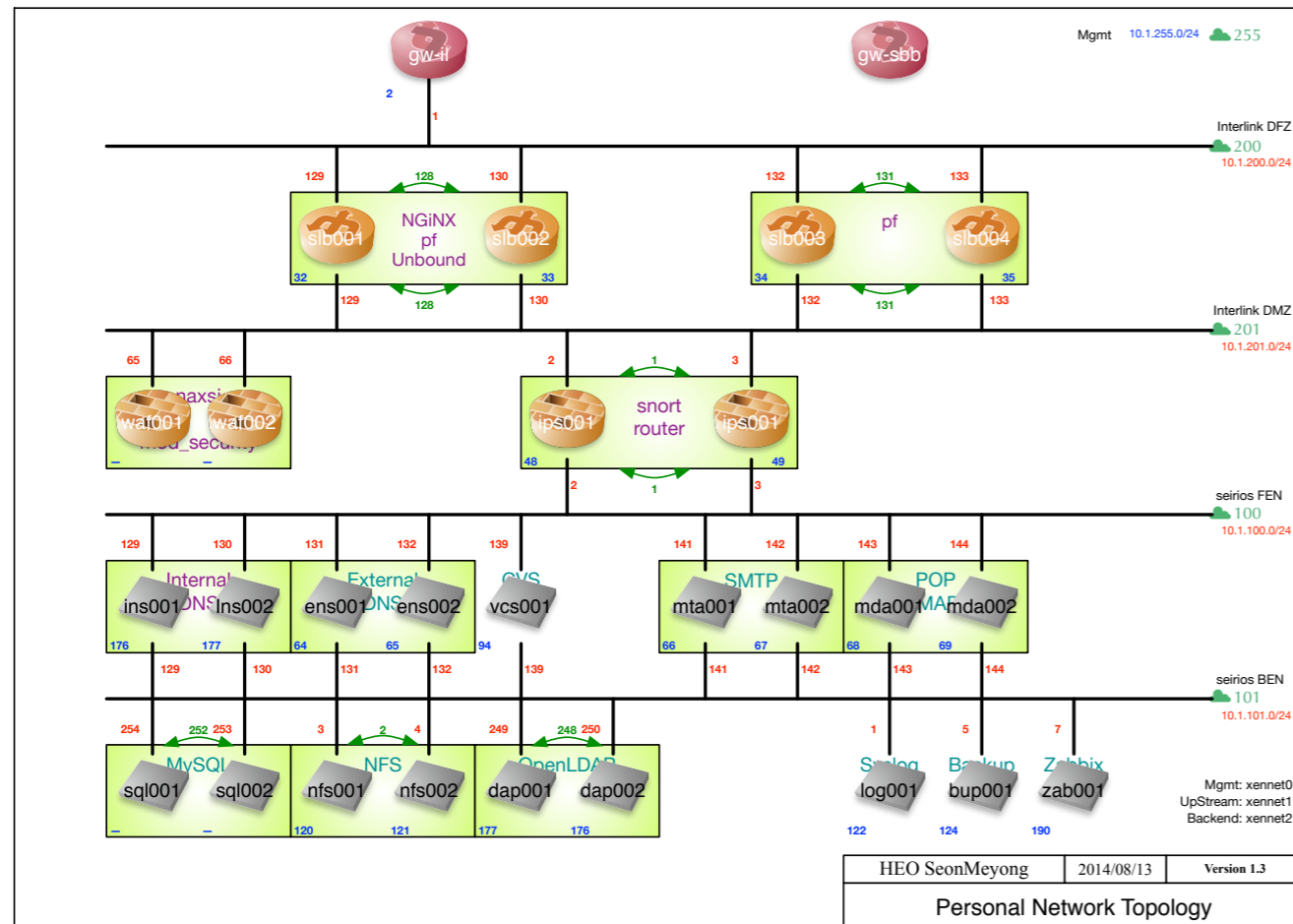


Printer  
&  
Consoles

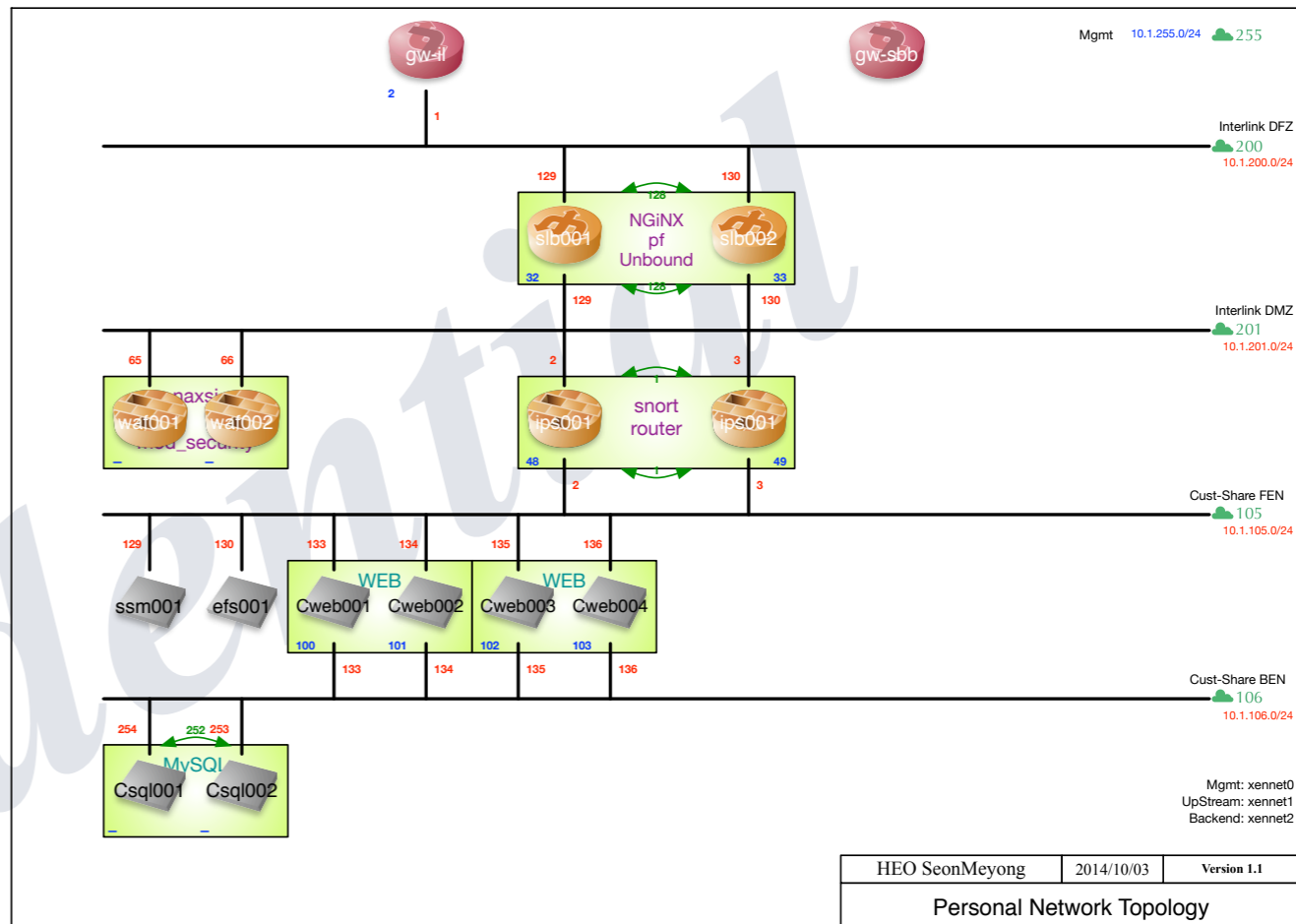
年間平均気温 25.3℃

# 我が家の環境 -2-

## 個人用(core)



## 知り合い収容用



システム構成図やサーバーのパラメータは  
全て管理している

現状のMonitoringはZabbixで行う  
RackTablesやRedmineなども仕掛けたい  
が手が回らない

ともあれ、古い機材をそろそろ止めたい

# 我が家の環境 -3-

## 📌 コンピュータルーム諸元

📌 広さ：3畳

## 📌 現在の消費電力

📌 コンピュータ系 定常状態で $12A + 12A = 24A$ ：24h/365days

📌 エアコン(8畳用) 24h/365days 冷房：設定温度 $18^{\circ}C$ 、室温 $25.3^{\circ}C$  (約 6.5A)

📌 月額電気料金（家庭分含む）：**約60,000(4月)～約75,000(7月)**

## 📌 なにやってんの？

📌 家庭内LAN環境

📌 仮想システム試験環境

📌 Xen/XenServer、完全冗長構成NAS実稼働試験、完全冗長DB環境試験

📌 個人及び知合の収容

📌 個人のWeb Pageや、いくつかの公開サービス

📌 個人で/28(IPv4)+/56(IPv6)、ドメイン4個+預かりドメイン3個

# Install

## VMの諸元

 CPU: 仮想CPU 1個

 Memory: 1024MB (UFSを利用するのでこのくらいでいい)

 HDD: 仮想HDD 20G + 仮想HDD 100G

 NIC: 仮想NIC x2

## 詳しい記事は以下を参照

 <http://www.seirios.org/seirios/dokuwiki/doku.php?id=os:freebsd:hast>

## 使った技術 . . . w

 HAST ・ CARP ・ devd ・ NFS

 CARPのStateがMASTERになったらHASTのroleをPrimaryにするためにdevd

# memo

📌 FreeBSD 10系におけるdevd.conf(CARP部分だけ)の例

```
notify 100 {  
    match "system" "CARP";  
    match "subsystem" "[0-9]+@[0-9a-z]+";  
    match "type" "(MASTER|BACKUP|INIT)";  
    action "/usr/local/sbin/carp-hast-switch $subsystem $type";  
};
```

📌 要するに、CARPのStateが変化したら、actionに記載されたコマンドを実行する。

📌 その時、引数に、I/F名と状態を渡す

これで、Stateが変化した時に、hastの挙動を変化させるためのscriptを呼び出せる。

# memo-2



carp-hast-switchを大幅に書き換えた(が、これ、使えない...)www

```
#!/bin/sh

#
# carp-hast-switch: shell script for change hast role when carp
# status is changed.
#
# Original script by Freddie Cash <fjwcash@gmail.com>
# Modified by Michael W. Lucas <mwlucas@BlackHelicopters.org>
# and Viktor Petersson <vpetersson@wireload.net>
# and HE0 SeonMeyong <seirios@seirios.org>
# Last modified 2015/11/10 HE0 SeonMeyong <seirios@seirios.org>

# ***WARNINGS***
# Need net.inet.carp.preempt=1 and same of advskew on carp.
# Currently HAST device must formatted by UFS
# ZFS code is implemented but not checked.
# This script is assumed to match the ZFS pool name and HAST resource name

#####
# Setting Variables and parse Arguments.

#DEBUG=1
SYSLOG_TAG="user.notice"
SYSLOG_TAG="carp-hast"

IF=${1%*}
VHID=${1#*@}
ACTION=$2

# Work around for boot time. devd execute this script before start hastd.
[ ! /bin/pgrep hastd ] && exit

case "${ACTION}" in
MASTER|BACKUP|INIT)
/usr/bin/logger -p ${SYSLOG_TAG} -t ${SYSLOG_TAG} \
"State Changed. I/F: ${IF} VHID: ${VHID} state: ${ACTION}"
;;
*)
/usr/bin/logger -p ${SYSLOG_TAG} -t ${SYSLOG_TAG} \
"FATAL: ${ACTION} is not yet implemented"
exit 1
;;
esac

#####
# Get resources.

HASTDEV="/sbin/hastctl dump all | /usr/bin/grep resource | /usr/bin/sed -e 's/^.*: \ *//'"
[ "x"$DEBUG != "x" ] && echo "HASTDEV = ${HASTDEV}"
[ -z "${HASTDEV}" ] && exit 0 # no hast device.

# get all carp interfaces
ifs="/sbin/ifconfig -l"
for i in $(ifs); do
no_of_carp="/sbin/ifconfig $i | /usr/bin/grep -c carp"
[ "x"$DEBUG != "x" ] && echo "Interface $i has $(no_of_carp) CARP configuration"
[ "${no_of_carp}" != "0" ] && carps="${carps} $i"
done
[ "x"$DEBUG != "x" ] && echo "CARP I/F = ${carps}"
[ -z "${carps}" ] && exit 0 # no carp I/F.

PREEMPTION="/sbin/sysctl net.inet.carp.preempt | /usr/bin/awk '{print $2}'"
[ "x"$DEBUG != "x" ] && echo "CARP preemption = ${PREEMPTION}"
[ "${PREEMPTION}" != "1" ] && exit 0 # No carp preemption. May cause failure.

#####
# Main.

case "${ACTION}" in
MASTER)
# make sure all carp is master.
if [ -n "${carps}" ]; then
for if in $(carps); do
vhid="/sbin/ifconfig ${if} | /usr/bin/grep carp | /usr/bin/awk '{print $3" $4}''
done
fi

for disk in $(HASTDEV); do
# If there is secondary worker process, it means that remote primary process is
# still running. We have to wait for it to terminate.
for i in `jot 30`; do
/bin/pgrep -f "hastd: ${disk} \\\(secondary\\)" >/dev/null 2>&1 || break
sleep 1
done
if pgrep -f "hastd: ${disk} \\\(secondary\\)" >/dev/null 2>&1; then
running after 30 seconds."
exit 1
fi

/usr/bin/logger -p ${SYSLOG_TAG} -t ${SYSLOG_TAG} \
"Role for HAST resources ${disk} switched to primary."
/sbin/hastctl role primary ${disk}
if [ $? -ne 0 ]; then
/usr/bin/logger -p ${SYSLOG_TAG} -t ${SYSLOG_TAG} \
"FATAL: Unable to change role to primary for
resource ${disk}."
exit 1
fi
done

# Wait for the /dev/hast/* devices to appear
for disk in $(HASTDEV); do
for loop in $(jot 120); do
[ -c "/dev/hast/${disk}" ] && break
sleep 0.5
done

if [ ! -c "/dev/hast/${disk}" ]; then
/usr/bin/logger -p ${SYSLOG_TAG} -t ${SYSLOG_TAG} \
"FATAL: GEOM provider /dev/hast/${disk} did not
appear."
exit 1
fi
FSFMT="file -bs /dev/hast/${disk}"
if [ $? -ne 0 ]; then
/usr/bin/logger -p ${SYSLOG_TAG} -t ${SYSLOG_TAG} \
"FATAL: /dev/hast/${disk} cannot define FS format."
exit 1
fi
FSFMT="echo ${FSFMT} | /usr/bin/awk '{print $1 \" \" $2}'"
case $(FSFMT) in
"Unix Fast")
/sbin/fsck -y -t ufs /dev/hast/${disk} >/dev/null 2>&1
if [ $? -ne 0 ]; then
/usr/bin/logger -p ${SYSLOG_TAG} -t ${SYSLOG_TAG} \
"FATAL: UFS fsck /dev/hast/${disk} failed."
exit 1
fi
/usr/bin/logger -p ${SYSLOG_TAG} -t ${SYSLOG_TAG} \
"FATAL: UFS fsck /dev/hast/${disk} failed."
fi
/usr/bin/logger -p ${SYSLOG_TAG} -t ${SYSLOG_TAG} \
"FATAL: UFS fsck /dev/hast/${disk} failed."
e_code="/sbin/mount /dev/hast/${disk} /hast/${disk} 2>&1"
if [ $? -ne 0 ]; then
/usr/bin/logger -p ${SYSLOG_TAG} -t ${SYSLOG_TAG} \
"FATAL: UFS mount for resource ${disk} failed: ${e_code}."
exit 1
fi
/usr/bin/logger -p ${SYSLOG_TAG} -t ${SYSLOG_TAG} \
"UFS /dev/hast/${disk} is mounted."
;;
*)
# If not UFS, Assume that filesystem is ZFS.
e_code="/sbin/zpool import -f ${disk} 2>&1"
if [ $? -ne 0 ]; then
/usr/bin/logger -p ${SYSLOG_TAG} -t ${SYSLOG_TAG} \
"FATAL: ZFS import for resource ${disk} failed: ${e_code}."
exit 1
fi
/usr/bin/logger -p ${SYSLOG_TAG} -t ${SYSLOG_TAG} \
"ZFS ${disk} is imported."
;;
esac
done
# NFS Service ( run nfsd and mountd. )
/usr/sbin/service rpcbind restart
/usr/sbin/service statd restart
/usr/sbin/service lockd restart
/usr/sbin/service nfsd restart
/usr/sbin/service mountd restart
/usr/bin/logger -p ${SYSLOG_TAG} -t ${SYSLOG_TAG} \
"NFS started."

# iSCSI service
# /etc/rc.d/iscsi_target start
;;

"BACKUP"|"INIT")
# make sure all carp is backup
if [ -n "${carps}" ]; then
for if in $(carps); do
vhid="/sbin/ifconfig ${if} | /usr/bin/grep carp | /usr/bin/awk '{print $3" $4}''
done
fi

for disk in $(HASTDEV); do
# If there is secondary worker process, it means that remote primary process is
# still running. We have to wait for it to terminate.
for i in `jot 30`; do
/bin/pgrep -f "hastd: ${disk} \\\(secondary\\)" >/dev/null 2>&1 || break
sleep 1
done
if pgrep -f "hastd: ${disk} \\\(secondary\\)" >/dev/null 2>&1; then
running after 30 seconds."
exit 1
fi

/usr/bin/logger -p ${SYSLOG_TAG} -t ${SYSLOG_TAG} \
"Role for HAST resources ${disk} switched to primary."
/sbin/hastctl role primary ${disk}
if [ $? -ne 0 ]; then
/usr/bin/logger -p ${SYSLOG_TAG} -t ${SYSLOG_TAG} \
"FATAL: Unable to change role to primary for
resource ${disk}."
exit 1
fi
done

# Wait for the /dev/hast/* devices to appear
for disk in $(HASTDEV); do
for loop in $(jot 120); do
[ -c "/dev/hast/${disk}" ] && break
sleep 0.5
done

if [ ! -c "/dev/hast/${disk}" ]; then
/usr/bin/logger -p ${SYSLOG_TAG} -t ${SYSLOG_TAG} \
"FATAL: GEOM provider /dev/hast/${disk} did not
appear."
exit 1
fi
FSFMT="file -bs /dev/hast/${disk}"
if [ $? -ne 0 ]; then
/usr/bin/logger -p ${SYSLOG_TAG} -t ${SYSLOG_TAG} \
"FATAL: /dev/hast/${disk} cannot define FS format."
exit 1
fi
FSFMT="echo ${FSFMT} | /usr/bin/awk '{print $1 \" \" $2}'"
case $(FSFMT) in
"Unix Fast")
/sbin/fsck -y -t ufs /dev/hast/${disk} >/dev/null 2>&1
if [ $? -ne 0 ]; then
/usr/bin/logger -p ${SYSLOG_TAG} -t ${SYSLOG_TAG} \
"FATAL: UFS fsck /dev/hast/${disk} failed."
exit 1
fi
/usr/bin/logger -p ${SYSLOG_TAG} -t ${SYSLOG_TAG} \
"FATAL: UFS fsck /dev/hast/${disk} failed."
fi
/usr/bin/logger -p ${SYSLOG_TAG} -t ${SYSLOG_TAG} \
"FATAL: UFS fsck /dev/hast/${disk} failed."
e_code="/sbin/mount /dev/hast/${disk} /hast/${disk} 2>&1"
if [ $? -ne 0 ]; then
/usr/bin/logger -p ${SYSLOG_TAG} -t ${SYSLOG_TAG} \
"FATAL: UFS mount for resource ${disk} failed: ${e_code}."
exit 1
fi
/usr/bin/logger -p ${SYSLOG_TAG} -t ${SYSLOG_TAG} \
"UFS /dev/hast/${disk} is mounted."
;;
*)
# If not UFS, Assume that filesystem is ZFS.
e_code="/sbin/zpool import -f ${disk} 2>&1"
if [ $? -ne 0 ]; then
/usr/bin/logger -p ${SYSLOG_TAG} -t ${SYSLOG_TAG} \
"FATAL: ZFS import for resource ${disk} failed: ${e_code}."
exit 1
fi
/usr/bin/logger -p ${SYSLOG_TAG} -t ${SYSLOG_TAG} \
"ZFS ${disk} is imported."
;;
esac
done
# NFS Service ( run nfsd and mountd. )
/usr/sbin/service rpcbind restart
/usr/sbin/service statd restart
/usr/sbin/service lockd restart
/usr/sbin/service nfsd restart
/usr/sbin/service mountd restart
/usr/bin/logger -p ${SYSLOG_TAG} -t ${SYSLOG_TAG} \
"NFS started."

# iSCSI service
# /etc/rc.d/iscsi_target start
;;

"BACKUP"|"INIT")
# make sure all carp is backup
if [ -n "${carps}" ]; then
for if in $(carps); do
vhid="/sbin/ifconfig ${if} | /usr/bin/grep carp | /usr/bin/awk '{print $3" $4}''
done
fi

for disk in $(HASTDEV); do
# If there is secondary worker process, it means that remote primary process is
# still running. We have to wait for it to terminate.
for i in `jot 30`; do
/bin/pgrep -f "hastd: ${disk} \\\(secondary\\)" >/dev/null 2>&1 || break
sleep 1
done
if pgrep -f "hastd: ${disk} \\\(secondary\\)" >/dev/null 2>&1; then
running after 30 seconds."
exit 1
fi

/usr/bin/logger -p ${SYSLOG_TAG} -t ${SYSLOG_TAG} \
"Role for HAST resources ${disk} switched to primary."
/sbin/hastctl role primary ${disk}
if [ $? -ne 0 ]; then
/usr/bin/logger -p ${SYSLOG_TAG} -t ${SYSLOG_TAG} \
"FATAL: Unable to change role to primary for
resource ${disk}."
exit 1
fi
done

# Wait for the /dev/hast/* devices to appear
for disk in $(HASTDEV); do
for loop in $(jot 120); do
[ -c "/dev/hast/${disk}" ] && break
sleep 0.5
done

if [ ! -c "/dev/hast/${disk}" ]; then
/usr/bin/logger -p ${SYSLOG_TAG} -t ${SYSLOG_TAG} \
"FATAL: GEOM provider /dev/hast/${disk} did not
appear."
exit 1
fi
FSFMT="file -bs /dev/hast/${disk}"
if [ $? -ne 0 ]; then
/usr/bin/logger -p ${SYSLOG_TAG} -t ${SYSLOG_TAG} \
"FATAL: /dev/hast/${disk} cannot define FS format."
exit 1
fi
FSFMT="echo ${FSFMT} | /usr/bin/awk '{print $1 \" \" $2}'"
case $(FSFMT) in
"Unix Fast")
/sbin/fsck -y -t ufs /dev/hast/${disk} >/dev/null 2>&1
if [ $? -ne 0 ]; then
/usr/bin/logger -p ${SYSLOG_TAG} -t ${SYSLOG_TAG} \
"FATAL: UFS fsck /dev/hast/${disk} failed."
exit 1
fi
/usr/bin/logger -p ${SYSLOG_TAG} -t ${SYSLOG_TAG} \
"FATAL: UFS fsck /dev/hast/${disk} failed."
fi
/usr/bin/logger -p ${SYSLOG_TAG} -t ${SYSLOG_TAG} \
"FATAL: UFS fsck /dev/hast/${disk} failed."
e_code="/sbin/mount /dev/hast/${disk} /hast/${disk} 2>&1"
if [ $? -ne 0 ]; then
/usr/bin/logger -p ${SYSLOG_TAG} -t ${SYSLOG_TAG} \
"FATAL: UFS mount for resource ${disk} failed: ${e_code}."
exit 1
fi
/usr/bin/logger -p ${SYSLOG_TAG} -t ${SYSLOG_TAG} \
"UFS /dev/hast/${disk} is mounted."
;;
*)
# If not UFS, Assume that filesystem is ZFS.
e_code="/sbin/zpool import -f ${disk} 2>&1"
if [ $? -ne 0 ]; then
/usr/bin/logger -p ${SYSLOG_TAG} -t ${SYSLOG_TAG} \
"FATAL: ZFS import for resource ${disk} failed: ${e_code}."
exit 1
fi
/usr/bin/logger -p ${SYSLOG_TAG} -t ${SYSLOG_TAG} \
"ZFS ${disk} is imported."
;;
esac
done
# NFS Service ( run nfsd and mountd. )
/usr/sbin/service rpcbind restart
/usr/sbin/service statd restart
/usr/sbin/service lockd restart
/usr/sbin/service nfsd restart
/usr/sbin/service mountd restart
/usr/bin/logger -p ${SYSLOG_TAG} -t ${SYSLOG_TAG} \
"NFS started."

# iSCSI service
# /etc/rc.d/iscsi_target start
;;

"BACKUP"|"INIT")
# make sure all carp is backup
if [ -n "${carps}" ]; then
for if in $(carps); do
vhid="/sbin/ifconfig ${if} | /usr/bin/grep carp | /usr/bin/awk '{print $3" $4}''
done
fi



for disk in $(HASTDEV); do
# If there is secondary worker process, it means that remote primary process is
# still running. We have to wait for it to terminate.
for i in `jot 30`; do
/bin/pgrep -f "hastd: ${disk} \\\(secondary\\)" >/dev/null 2>&1 || break
sleep 1
done
if pgrep -f "hastd: ${disk} \\\(secondary\\)" >/dev/null 2>&1; then
running after 30 seconds."
exit 1
fi

/usr/bin/logger -p ${SYSLOG_TAG} -t ${SYSLOG_TAG} \
"Role for HAST resources ${disk} switched to primary."
/sbin/hastctl role primary ${disk}
if [ $? -ne 0 ]; then
/usr/bin/logger -p ${SYSLOG_TAG} -
```

# Problems

# さて本題

## 問題点は

-  FreeBSDがBootする際の挙動
-  NFSが(この用途には)使えない

# Boot時の挙動問題

## 起動時の挙動(必要部分だけ)

1.devdを起動

devdがNIC設定より前に起動される

→NIC設定(CARP設定)時にscriptが呼ばれる orz..

2.NICを設定

3.rpc関連を起動

4.NFS Client関連の起動

hastdが起動された時にはすでにNICは設定済み

5.hastd起動

→scriptは呼ばれない

→結果、hastd起動後にroleを設定できない

6.NFS Server関連の起動

hastdがprimaryになっていないのにExport

→されるはずがない

# Boot時の挙動問題(解決策)

- 📌 システム起動に関しては手をつけない
  - 📌 rc.confとかrc.conf.localとかでFakeするのは、保守上ダメ
- 📌 対応策(案)
  - 📌 carp-hast-switchをいじる
    - 📌 hastedの起動状態の確認を行う（起動してなければ何もしないようにする）
  - 📌 /etc/rc.localで、最後にcarp-hast-switchを呼び出す
  - 📌 CARP Stateの確認が必要

再起動問題さえなければ、現状でちゃんとHASTは動いています。

しかも、かなり安定しています。

HASTを止めてmountすれば、ちゃんとFilesystemとして見えます。

でも、ここをちゃんとしないと、**SplitBrain**になります

# NFSが使えない -1-

- 📌 使えそうなNetwork File Systemはあるか？
  - 📌 FreeBSDで使えそうなものはほとんどない。
  - 📌 ClientにCentOS/Ubuntu/NetBSD/OpenBSDがあるので、そもそも制限がきつい
  - 📌 実質NFSだけじゃね？、あえて言うならSAMBA???、(嫌～)
- 📌 NFS : Network File System
  - 📌 NFS: RFC1094 (1989/03)
  - 📌 NFS version 3: RFC 1813 (1995/06)
  - 📌 NFS version 4: RFC 3530 (2003/04)
    - 📌 RFC3010(Obsolete): (2000/12)
- 📌 つまるところ、古い…

# NFSが使えない -2-

📌 いや、普通に使う分には使えます。

📌 NFS v4は認証関係でもものすごく面倒そうだったので2035年問題があるのを承知で  
NFSv3にしました

📌 別に普通にNFS鯖立てて、NFS暗蟻作れば普通にFilesystem共有できますよ。

**駄菓子菓子**

📌 完全冗長にはできなかったのです。

# NFSが使えない -3-

- 📌 NFS ServerはNFSでの接続ポイントをCARPで供給している
  - 📌 NFS ServerのIPアドレスをNs1/Ns2とし、NFSのアドレスをNsCとする
  - 📌 NFS ClientをNcとする
  - 📌 初期状態では、Ns1がNsCを持っていて、NcがNFS mountしているとする

## 📌 問題発生の流れ

1. Ns1がshutdown
2. NsCがNs2に移り、Ns2はHAST Primaryになり、NFS関連が起動する
3. NcはNsCを通じてNs2のStorageにアクセスしようとする
4. いつまでたってもアクセスできず、はまる

# NFSが使えない -4-

📌 なにが起きているのか？

📌 NFSでFilesystemを公開しているのはmountd

📌 mountdは、NFS Clientを記録している

📌 /var/db/mountdtabにClientのIP Addressを記録している

📌 NFS Server #1と#2は、当然、HASTのVolumeしか同期していない

📌 したがって、当然mountdtabにはClientのIP Addressは記録されていない

📌 結論

📌 NFS Server#2はNFS Clientからの接続を認めない！orz..

# NFSが使えない -5-

- 📌 まず、そもそものArchitectureを考え直す
  - 📌 HASTは、Filesystemの完全な同期だから、NFS Client側のinode cacheに対して耐性があるはず
  - 📌 耐性があるなら、NFSのCacheは忘れてもなんとかなるか..
- 📌 じゃあ、どうすりゃいいのよ？
  - 📌 まあ、取り急ぎで言うなら、/var/db/mountdtabを同期する手を考える
  - 📌 でもどうしよう？rsync?ないよなあ。
  - 📌 HAST???? 動く気がしないでもないが、えーと...
  - 📌 automountで凌ぐ？
  - 📌 動く気がしない…。いや、動いてくれそうな気もするんだけど...

📌 何らかの知恵求む